Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1987-09

Application of a database system for inventory management of classified material.

Lim, Rolando Mateo

http://hdl.handle.net/10945/22636

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

Y
SCHOOL
, CALIFORNIA 93943-5002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

APPLICATION OF A DATABASE SYSTEM FOR
INVENTORY MANAGEMENT OF CLASSIFIED MATERIAL

by

Rolando Mateo Lim

September 1987

Thesis Advisor: Norman R. Lyons

Approved for public release; distribution is unlimited

T234289

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is Unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) Code 54	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Application of a Database System for Inventory Management of Classified Material (u)

12 PERSONAL AUTHOR(S)
Lim, Rolando Mateo

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	September 1987	80

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Database application shipboard inventory management

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis presents a database application in the area of inventory management of classified material. The system is designed for shipboard application, but can be used in the shore establishment as well.

A simple database using dBASE III PLUS is implemented on an IBM PC. It is designed for users who have very little computer experience.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Norman R. Lyons	22b TELEPHONE (Include Area Code) (408) 646-2666 22c OFFICE SYMBOL Code 54Lb

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Approved for public release; distribution is unlimited.

Application of a Database System for Inventory Management of Classified Material

by

Rolando Mateo Lim
Lieutenant, United States Navy
B.S., New Hampshire College, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

ABSTRACT

This thesis presents a database application in the area of inventory management of classified material. The system is designed for shipboard application, but can be used in the shore establishment as well.

A simple database using dBASE III PLUS is implemented on an IBM PC. It is designed for users who have very little computer experience.

Thesis
263560
c.1

TABLE OF CONTENTS

I. INTRODUCTION 7

II. BACKGROUND 9
A. PROBLEM DESCRIPTION AND ORGANIZATION . . . 9
B. GENERAL OVERVIEW OF A DATABASE SYSTEM . . . 11
1. Introduction 11
2. Architecture of a Relational Database . 11
3. Logical vs Physical Models of Databases 14

III. SYSTEM ANALYSIS 16
A. INTRODUCTION 16
B. DESCRIPTION OF THE CURRENT SYSTEM 16
C. PROPOSED IMPROVEMENT OVER THE CURRENT SYSTEM 18
D. JUSTIFICATION OF A COMPUTERIZED SOLUTION . 20
E. SYSTEM REQUIREMENTS 20
1. Classes of System Operations 21
2. Expected Volume of Data 22
3. Kinds of Queries to be Answered 22
4. Relative Update and Retrieval 22
F. INPUT REQUIREMENT 23
G. OUTPUT REQUIREMENT 23

IV. RELATIONAL DESIGN 24
A. MODELING 24
1. Record Structure 24
2. Record Relationship Diagram 25
B. DATA DICTIONARY 27

V. IMPLEMENTATION 31
A. INTRODUCTION 31

B. SHIPBOARD IMPLEMENTATION 31
1. Software Requirement for System Implementation 32
2. Hardware Requirement for System Implementation 32
C. DATABASE ADMINISTRATION 32

VI. SAMPLE PROGRAM PROCESSING 34
A. MAINPROG 35
B. HELP 36
C. UPDATE DATABASE 36
1. Select Option "1" to Insert Pubs Record 37
D. REPORT GENERATOR 37
1. Select Option 1 to Get Allowance Listing 38
E. ADHOC REQUEST 38

VII. CONCLUSION. 39

APPENDIX A: PROGRAM STRUCTURE CHART 40
APPENDIX B: PROGRAM 41
LIST OF REFERENCES. 78
INITIAL DISTRIBUTION LIST 79

ACKNOWLEDGMENT

I would like to give my deepest appreciation to the Navy for investing in my graduate education.

Furthermore, my sincere thanks to Professor Norman R. Lyons, whose willing support aided in the completion of this thesis.

In addition, my appreciation to LCDR James R. Duke for his significant comments and useful suggestions.

Finally, a special thanks to my wife, Luz for her encouragement and support throughout my graduate study.

I. INTRODUCTION

Inventory management of classified material is of vital importance to the armed forces of the United States. There has been an increasing threat to military secrets, which prompted considerable changes in the area of inventory management of classified material. Much of this increase can be attributed to the growing complexity of intelligence gathering activities of foreign nations; however, this thesis will examine only the problem as it relates to inventory management of classified publications held by ships in the U.S. Navy. For a commanding officer of a ship, it is imperative that inventories of classified material be performed accurately and effectively. To achieve this goal, the ship's security manager needs a modern database management system that will provide an accurate inventory listing and quick access to information when required.

A modern database management system permits greater flexibility in meeting information requirements, faster response time, better data security, and easier access to stored information than earlier software systems although at a greater cost. However, these benefits (along with greater accuracy and increase productivity) strongly justify the necessary large capital and manpower investments. Perhaps the most exciting development of such systems is the number of available, easy to use query-type languages which permit novice users to create, update, maintain, and extract information from their own files.

During the database development process, users' requirements are translated into system requirements

using application programs or the Database Management System (DBMS) itself. The normal form concepts of relational database models are applied to develop a database for the inventory management of classified material to be used onboard Naval ships.

This thesis focuses on a database application for maintaning inventories for the Secret custodian. While this application could be used by both Top Secret and Secret custodians, it is the intention of this thesis to concentrate on the Secret custodian's inventories system needs. The Relational Model was applied to design this inventory database system.

II. BACKGROUND

A. PROBLEM DESCRIPTION AND ORGANIZATION

As a result of the well publicized spy trial of former Radioman Senior Chief Whitworth in 1984, the Navy launched a more extensive campaign to improve accountability of classified materials. Stringent standard procedures are published in the Chief of Naval Operations Instruction 5510.1, Fleet and Type Commanders directives. This guidance is incorporated into a ship's own instructions. While these voluminous instructions attempt to cover all contingencies, the problem encountered by the security managers onboard ships was one of trying to manage a large body of information with very little assistance in the area of inventory management.

One officer is normally assigned as security manager on each Navy ship. One of his many duties is to ensure that all classified materials onboard are properly accounted for. The security manager normally controls any access to Top Secret publications. In order to minimize this access, the security manager physically must conduct inventories and control access.

The immediate assistant to the security manager with regards to inventory management of classified material is the Secret custodian. As the title implies the Secret custodian is responsible for all Secret and Confidential materials held onboard ship. The Secret custodian is normally a junior officer with little or no experience in the management of classified material.

There are three distinct groups of publications: "Naval Warface and Allied Tactical", "Intelligence", and "Secret" which contains all Secret materials not

relating to the first or second group. To properly manage these inventories, the secret custodian is assisted by three enlisted personnel as sub-custodians. The task of the sub-custodians is to help the secret custodian maintain an accurate inventory of the publications and make required updates whenever necessary. The sub-custodians are assigned lockers (safes), for which he or she is responsible. Each locker (safe) contains Secret and Confidential materials, mostly in binded publications and micro-fiche. Files of Secret messages and notes that belong to the commanding officer and other officers from the unit are also kept in the locker. The Navy has not adopted a standard format for inventorying these classified materials at this time, which leads to disorganization and resultant difficulties in determining current inventory in a timely manner .

Although the task of the secret custodian is considered very important and of high military interest, it is but a collateral duty which usually takes second place to primary duties. Publication updates and deletions are typically put off until there is a slowdown in the operations tempo. This is acceptable for the most part; however, this potential lack of immediate attention could cause a loss or misplacement of classified material and result in damage to the Navy's mission and at least a loss of face to the men involved.

The problem is critical both in large and small ships. In the first case, there are a large number of publications based on ship type. For example, a carrier will have an enormous amount of classified publications to account for and may have numerous sub-custodians. In the second case, the limited number of people that

serve on the ship makes the assignment of additional duties much more difficult.

B. GENERAL OVERVIEW OF A DATABASE SYSTEM

1. Introduction

It was not until the late 1950s that computers were successfully used for data management. These computers were able to process data quickly and in great quantities; however, these computers were still unsophisticated (limited storage, manipulation, and retrieval capabilities). In the middle 1960s, computer architecture was radically changed. A large increase in the size of computer memory and the introduction of operating systems made multi-programming possible, which means two or more jobs can be done concurrently.

Along with multi-programming came the capacity to do on-line or single transaction processing. Rather than process large volumes of data sequentially, it has become economically feasible to access specific information from the computer. By the late 1960s, more sophisticated methods of storing and retrieving data were incorporated into computer software that gave birth to database management as we know it today.

2. Architecture of a Relational Database

Relational database software organizes data into simple tables called relations. Such software allows users to create reports by drawing data from multiple tables. This provides a simplicity and flexibility of use not easily obtained in non-relational database systems.

The concept of relational database was first publicly proposed by Dr. E. F. Codd in a paper entitled " A Relational Model of Data for Large Shared Data Banks" [Ref. 1]. Codd suggested a theoretical method for representing data based concepts developed in the

field of relational mathematics. Until the early 1980s the ideas were only of theoretical interest. Now there are several database management systems available for microcomputers based on these relational concepts. Some of these include dBASE III, dBASE III PLUS, R:Base, Knowledge Man, Ingres, and ORACLE.

Relational database principles are based on relational theory in mathematics. Many of the terms used in relational database management software are used in relational mathematics. For example, a two-dimensional table is called a relation, where one dimension is called a row and the other a column.

In a relational database, each row is called a tuple. A tuple contains data items or entities. The more familiar term for a tuple in data processing is a record. In a relational database, a collection of related tuples or records is called a file. Figure 2.1 illustrates a portion of a relation called SAILOR that might be found in the database of a Naval Facility.

Columns in a relation are called attributes which contain data describing the relations. In Figure 2.1 the attributes are NAME, SSN, DEPT, AGE, and PRD (rotation date). Because there are five attributes in each tuple, they are called five-tuples. A relation that contains five attributes is called a relation of degree five. A relation that has six columns, or six attributes, is said to be of degree six, one with seven attributes, degree seven, and so on.

The relation in Figure 2.1 is also referred to as a flat file. A flat file is called flat because it has only two dimensions (rows and columns). Also, a flat file has no repeating groups or fields. Each record within a flat file has the same number of fields, even if some of them are blank.

All files within a relational database must be flat files. In data processing terminology, this means that all records within one file must have the same number of fields, even though some of the fields may be blank for some records. It also means that any file within a relational database can be visualized as a table with a fixed number of columns. Each column is a

NAME	SSN	DEPT	AGE	PRD
Adams	111-11-2111	04	22	8/86
Brown	987-65-4321	05	30	9/87
Carson	222-11-7687	05	32	6/86
Carson	128-55-8765	05	30	9/87
Davis	229-77-0090	03	19	10/87
Forbes	556-33-7654	04	21	9/86
Kelley	112-87-1765	06	20	8/86
Lee	565-08-7611	06	25	9/87
Pinoy	573-12-1187	04	27	11/87
Roberts	126-28-7653	06	18	12/87

Figure 2.1 A Sailor Relation.

fixed number of columns. Each column is a field containing data that either uniquely identifies the record or describes the record. If the data uniquely identify the record, then that attribute is the key. For example, in Figure 2.1, SSN is the key and NAME, DEPT, AGE, and PRD are all attributes that describe a particular sailor.

3. Logical vs Physical Models of Databases

In relational database design a major distinction is made between the logical model and physical model. A logical model is an abstract representation of the entities in the database and the relationships between those entities. Figure 2.2 illustrates a logical model of sailors assigned in divisions of departments. The entities are shown as boxes and the relationships as a diamond. The entities are called SAILOR and DIVISION and stand for sailors and divisions of the departments. Records would be kept regarding each sailor and each division of a department. SAILOR/DIVISION represents the relation we would describe as: A sailor is assigned to a division within a department. Records will be kept regarding the sailors assigned to a division.

Figure 2.2 Logical Model.

The next stage of the database design is the construction of the physical model. A physical model is the description of the database in terms of a specific database. Figure 2.3 is a physical model of the logical model in Figure 2.2.

A logical model is made without concern for the particular structure or restriction of the database management software or computer hardware that will process the database. A physical model must be designed to work within the constraints of the particular hardware and software used, such as amount of storage

available, maximum file size, naming conventions, and available commands [Ref. 2].

```
DATABASE USS SHIPS
 RELATION SAILOR

   01          NAME          C     25
   02          SSN           C      7
    :
    :

 RELATION DIVISION
   01          NAME          C     25
   02          POSITION      C     20
   03          WORKSTATION   C     20
    :
 RELATION SAILOR/DIVISION
   01          SSN           N      7
   02          NAME          C     25
    :
```

Figure 2.3 Physical Model.

III. SYSTEM ANALYSIS

A. INTRODUCTION

System analysis is the process of gathering, interpreting, and using facts to improve an existing system through better procedures and techniques. In other words, analysis specifies what the system must do and should do.

The first step of the analysis phase is a system study which includes the accumulation of information relating to the existing system's capabilities and deficiencies. This information may lead to a need for new system development or a major improvement to an existing system.

According to J. A. Senn [Ref. 3] the reasons for a project initiation are:

a. Greater processing speed
b. Better accuracy and improved consistency
c. Faster information retrieval
d. Reduced cost
e. Better security

In general, all of the above reasons should be satisfied by the new system; however, it is for the organization to decide if the new system start is necessary or required.

B. DESCRIPTION OF THE CURRENT SYSTEM

The present system on most ships is graphically described in Figure 3.1 using a Yourdon's data flow diagram. The circle in the diagram represents the process being done at the time. An arrow represents the flow of the data, information, or materials. The two parallel lines represent a data bank or database.

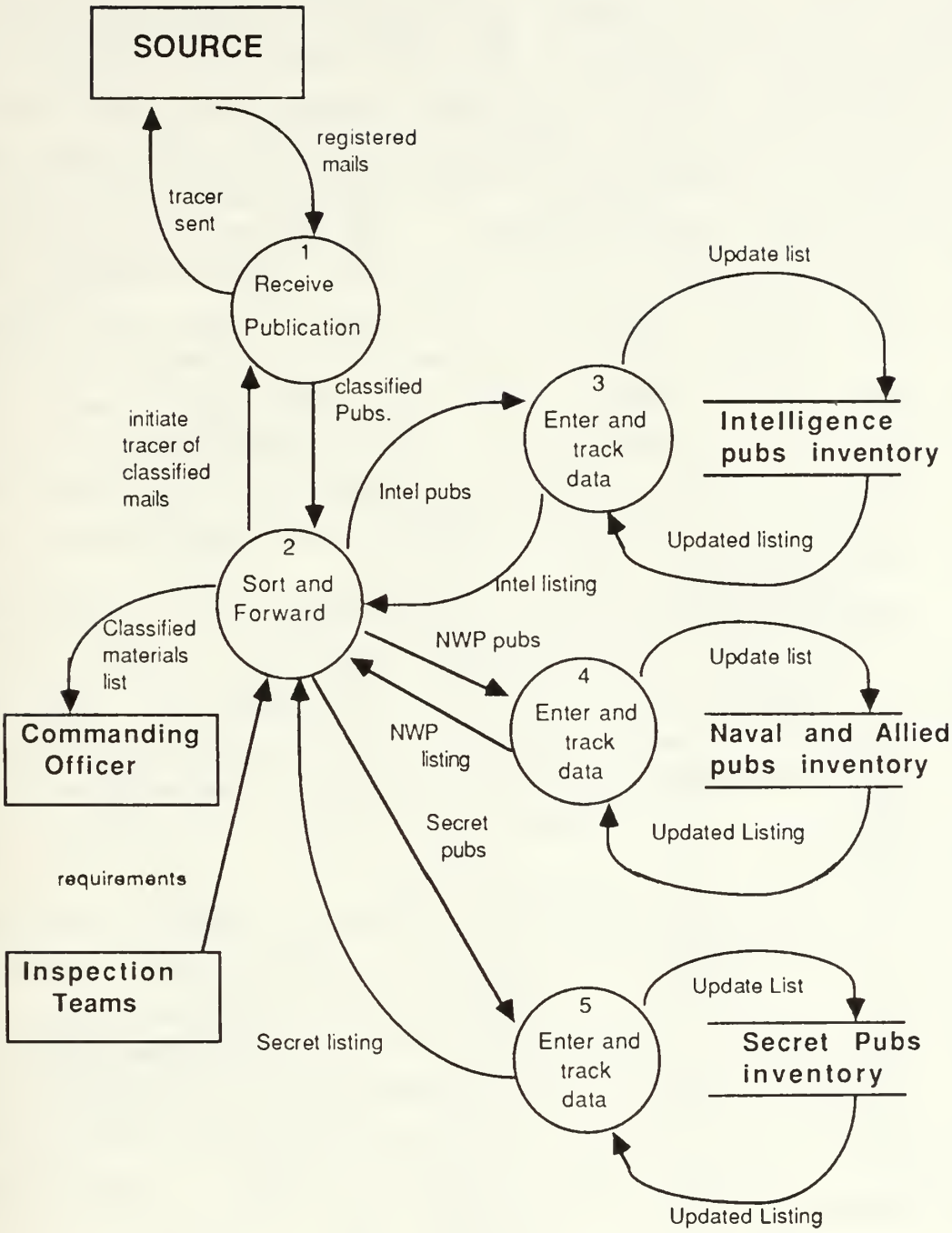


Figure 3.1 Present System on Most Ships.

The mail which contains classified material is delivered to the ship by registered mail. The ship's yeoman receives, signs, and distribute the mail to the appropriate personnel. In the case of Secret and Confidential materials, the secret custodian will receive and sign for custody. The secret custodian then sorts and forwards the publications (registered mail) to the proper sub-custodians. The sub-custodians check the publications for accuracy, completeness, and include changes whenever necessary. Each publication will be logged and included in the inventory of the sub-custodian's safe. Publications are kept and organized by document type. Updates and deletions are not reflected until update listings are available. The publication listings are manually typed and reports are not often available when required.

The sub-custodians are responsible for the physical safety of the publications that are under their controls. Updates and deletions are manually made and indicated on the publications. Publications earmarked for destruction are normally grouped with the rest of the expired publications for future shredding and incineration.

Each sub-custodian is required to have a current listing of his safe's inventory. An accurate inventory listing is normally required semi-annually by the security manager for the purpose of reporting status to the executive officer and commanding officer. Inventory listings are also required during security inspections, training inspections, and normal operational deployments.

C. PROPOSED IMPROVEMENT OVER THE CURRENT SYSTEM

The proposed system is graphically described in Figure 3.2. The yeoman's job in process 1 will still be

the same; however, the secret custodian will have only one sub-custodian for all the lockers (safes). Each piece of classified material will be assigned a control number by the secret custodian. Control numbers will be used to uniquely identify each publication or classified materials belonging to the secret custodian. The sub-custodian will do the data entry and all the previously described tasks currently performed manually.

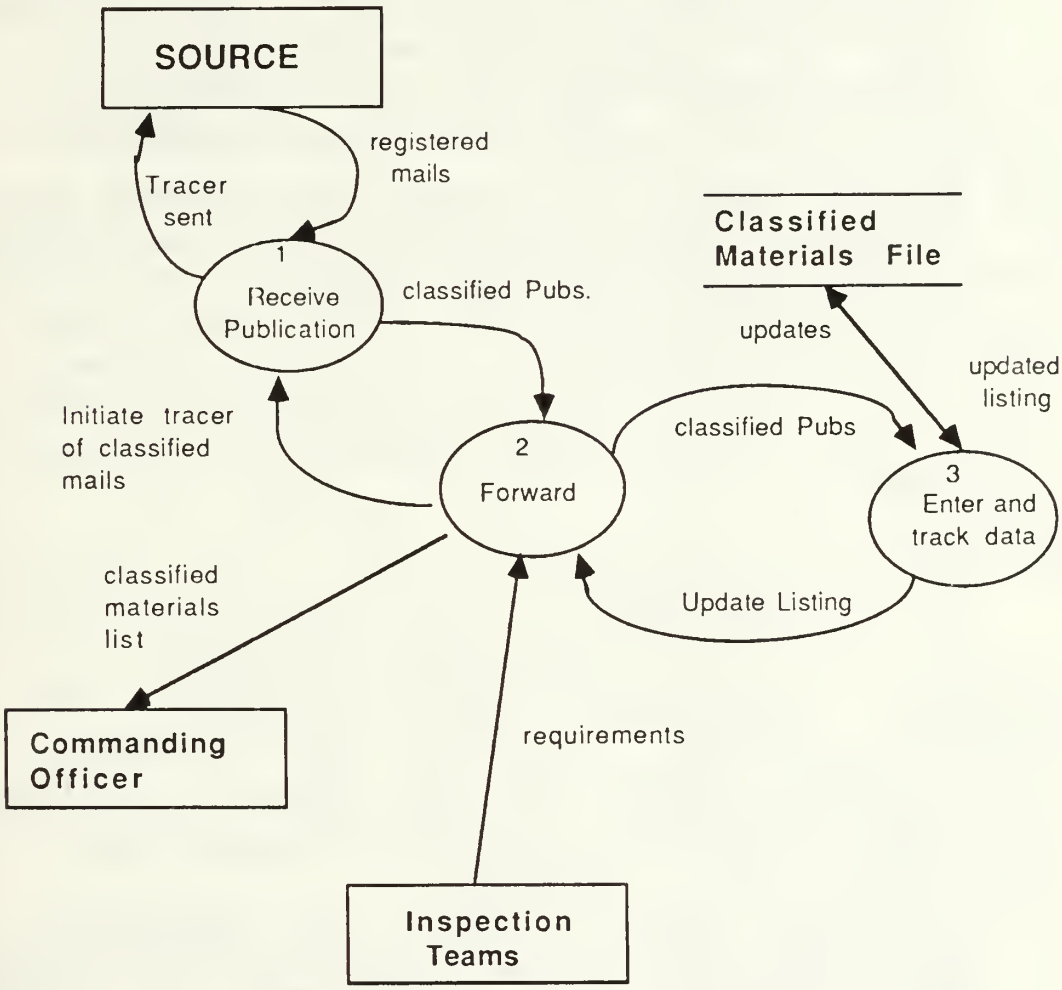


Figure 3.2 Proposed System.

D. JUSTIFICATION OF A COMPUTERIZED SOLUTION

The following improvements can be achieved by computerized solution or automation:

a. The processing speed will be much faster and eventually simpler. One person should be able to perform the whole task effectively.

b. Improved accuracy in data retrieval can be achieved by eliminating the possibility of information or data omission.

c. The retrieval and listing of inventory information will be much faster, easier, and real time.

d. In a military environment, the cost of the manually provided functions are not easily estimated. Therefore, cost reduction as a requirement cannot be considered an improvement to the existing system, except in terms of man-hours and the number of personnel required to do the job. On the other hand, the monetary cost of installing a micro-computer system is considered insignificant since the proposed system will be designed to be supported by a stand alone micro-computer.

e. The level of security will be limited to a few system traps and program passwords. However, the physical security applied to publications will be extended to the program software.

The previously mentioned improvements lead to the decision that this project is a good candidate for a new system development. It satisfies the requirements for project initiation as stated in the beginning of this chapter.

E. SYSTEM REQUIREMENTS

System requirements were determined through a series of interviews with users at both sea and shore commands. Incorporating users' and management needs, the proposed system must be able to satisfy the following requirements so that the previously defined improvements can be achieved:

a. It must be able to store any information about classified material inventory currently held by Navy ships. Actual storage may be different from organizations to organizations; however, modifications in storage maybe necessary to avoid data duplication.

b. It must be able to provide any stored information or combination of stored information upon request.

c. It must be easy to use. Personnel with very little knowledge about computers should be able to operate the system satisfactory.

d. It must be supportable on a stand alone micro-computer. Security of data will be managed effectively with a stand alone micro-computer system.

1. Classes of System Operations

The system will provide the following four classes of operations:

a. Modification of the Existing Data

The user of the system should be able to insert a new record, to delete an existing one, and to modify any record in the supporting system files.

b. Locker Allocation

The system should be able to provide different types of publication allocation. For example, certain types of publication will go on certain locker(s) and the location of the locker(s). The criteria used for each type of publication allocation will be described with the corresponding application program.

c. Production of Reports

A number of application programs will support the function of retrieving necessary information about publications from the database and producing the appropriate lists and reports.

d. Assistance Operations

Although the above operations can satisfy the request of the system, an additional function related to the tracking of modifications that have been done to the supporting system files is necessary. This audit history will be contained in file "TRACER" in which any modifications of the supported files will be recorded with the time, date, and user name. For security reasons, this file should not be accessible through the main menu or any sub-menu. Also, system

access security should be provided at system start-up through a password control system.

A "USERID" file will contain users' names and corresponding passwords. A second "USERLOG" file will accumulate the information of all transactions done by the user of the system. Neither file will be accessible through the menu system.

2. Expected Volume of Data

The volume of data should average approximately 1600 binded publications, 1000 micro-fiche, 300 binded Secret notes and several item of other classified material, both in binded publication and loose messages received during operations.

3. Kinds of Queries to be Answered

The database would be entered through the main menu. The user could select a file for querying or updating a specific file. The queries answered through the database include the following:

a. Listing of all publications by locker numbers.
b. Location of each or a certain publication.
c. The sub-custodian of this publications.
d. The update of a certain or many publications.
e. The total number of Secret and Confidential material.
f. The basic allowance list vs current publication available onboard.
g. The total number of publications in each locker.
h. The destruction record of the last two years.

4. Relative Update and Retrieval

The database shall be updated monthly or more frequently as the custodian feels necessary. The database retrieval activities shall be performed periodically as follows:

a. Quarterly for inventory purposes.
b. As requested or deemed necessary by the custodian or higher authority.

Additionally, ad hoc queries shall be performed when desired by the security manager and commanding officer.

F. INPUT REQUIREMENT

The input requirement are the system requirements and project plan. These requirements were determined through publication instructions, and users' needs through interviews with previous secret custodians and staff personnel whose job involved inventory management of classified publications.

G. OUTPUT REQUIREMENT

A similar process of acquiring information was conducted in formulating output requirements. Output requirements may be altered by the next user as they apply to the present regulations.

IV. RELATIONAL DESIGN

Logical design is the first major task in the design of a database-oriented system. This task will be to define the database structure, which is a two-phased process. First, the logical structure of the database is developed; second, the physical design of the database structure is developed.

A. MODELING

The essence of database design is the representation of record relationships.The relationships can be specified using a data structure diagram (DSD), commonly called a Bachmann diagram. The relationships are identified intuitively. Design was conducted on the potential relationships among records that have been defined for the system.

1. Record Structure

In order to satisfy the user's requirements, a number of records were derived from a publication listing held by the secret custodian. Because of military security, bogus data and other information were entered to the record model instead of real data. From the information in the publication record, a relation model of the system was built. In this thesis, the following records were generated:

a. PUBS (Ctrl_No, S_Title, Locker, Class, Update, Sub_Cust)

Ctrl_No: control number

S_Title: short title

Locker: locker number

Class: classification

Update: last update done

Sub_Cust: sub-custodian

b. PUBSINFO (S_Title, L_Title, Allowance, Onboard, Doc_Type)
   S_Title: short title
   L_Title: long title
   Allowance: allowance for ship
   Onboard: onboard publications
   Doc_Type: document type
c. DESTRUCT (Ctrl_No, S_Title, D_Destruct)
   Ctrl_No: control number
   S_Title: Short title
   D_Destruct: date of destruction
d. USERID (Name, Password)
   Name: name of user
   Password: password of user
e. USERLOG (Logdate, Logtime, Name, Task, Progname)
   Logdate: logged date
   Logtime: logged time
   Name: name of user
   Task: task performed
   Progname: program names accessed

Underlined field(s) in the record are the key. Precise information about each field is in the Data Dictionary in Section B of this chapter.

2. Record Relationship Diagram

DSD was used for the representation of record relationships. Each relation described in the above section and Figure 4.1 is in third normal form (3NF). This means that each tuple or record of each relation consists of a primary key value that identifies some entity, together with a set of mutually independent values that described the record in some way. For

consists of a primary key value that identifies some entity, together with a set of mutually independent values that described the record in some way. For example, the key in the PUBSINFO relation describe a unique tuple within that relation. The single or double arrow notation is used to express relationship between records (one-to-one, and many-to-many relationships) shown in Figure 4.1.

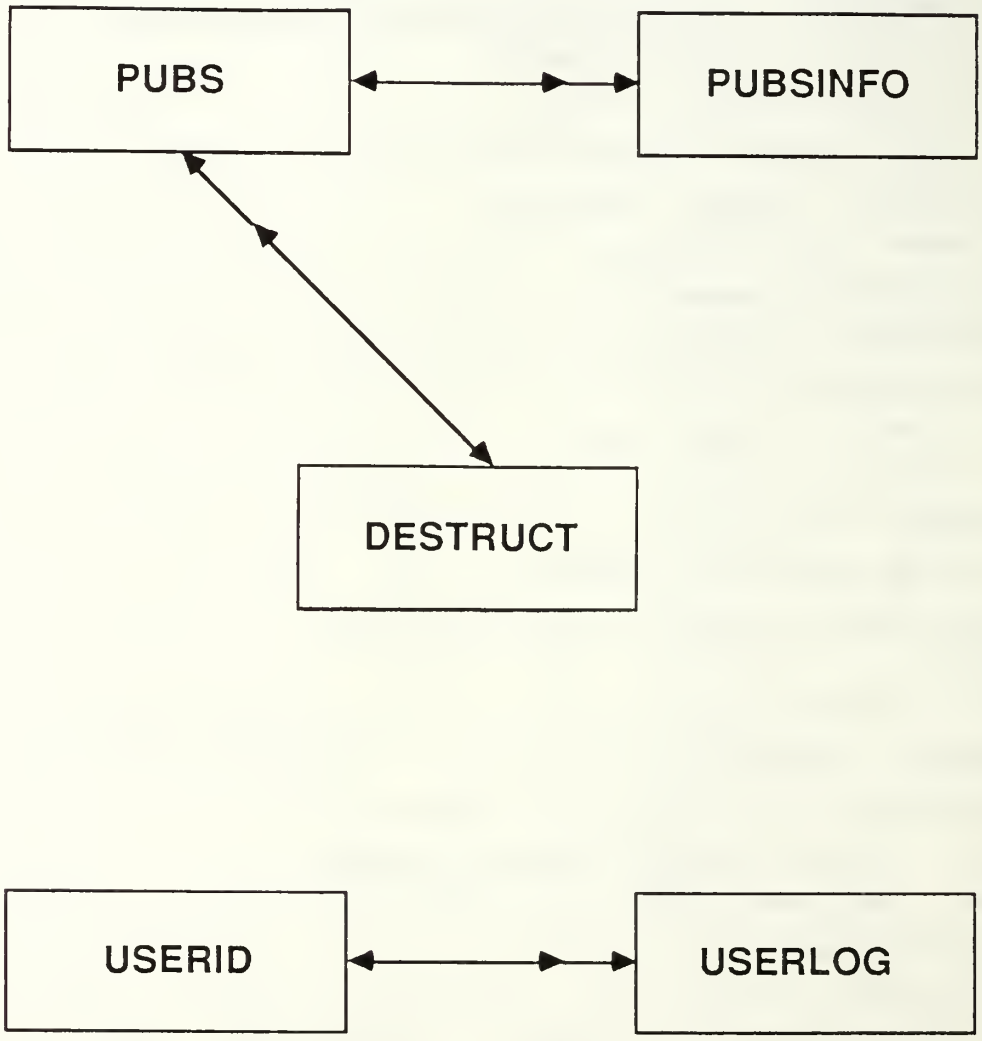


Figure 4.1 Data Structure Diagram (DSD).

B. DATA DICTIONARY

The data dictionary is the collection of correct information about the words and terms used by the organization in order to describe its data. The term data dictionary is used to indicate a set of files or a database. The data dictionary for this thesis is:

Field Name: Ctrl_No
Format: character
Width: 5
Allowable value: less than 50,000
Description: Numeric, character plus numeric

Field Name: S_Title
Format: character
Width: 20
Allowable value: short title, originator's name
Description: short title of publications

Field name: L_Title
Format: character
Width: 40
Allowable value: long title, subject name
Description: long title of publications

Field name: Locker
Format: character
Width: 1
Allowable value: characters, numbers
Description: locker id (i.e., locker 2)

Field name: Class
Format: character
Width: 9

Allowable value: SECRET, CONF, UNCLASS
Description: publication classification

Field name: Update
Format: date
Width: 8
Allowable value: MM/DD/YY
Description: date of last update

Field name: Sub_Cust
Format: character
Width: 12
Allowable value: last name, initials, rank
Description: last name and rank of person

Field name: Allowance
Format: numeric
Width: 4
Allowable value: 0 to 9999
Description: number of publication required

Field name: Onboard
Format: numeric
Width: 4
Allowable value: 0 to 9999
Description: number of publication onboard

Field name: Doc_Type
Format: character
Width: 7
Allowable value: NTP/ATP, SECRET, INTEL
Description: publication by document type

Field name: D_Destruct
Format: date

Width: 8
Allowable value: MM/DD/YY
Description: destruction date

Field name: Name
Format: character
Width: 12
Allowable value: last name and rank
Description: sub-custodian's name and rank

Field name: Password
Fromat: character
Width: 8
Allowable value: numeric, alpha-numeric, characters
Description: user's access code

Field name: Task
Format: character
Width: 10
Allowable value: INSERTION, LIST, and other tasks
Description: program task performed by each user

Field name: Progname
Format: character
Width: 10
Allowable value: ADD_PUBS, and other program names
Description: program names in this thesis

Field name: Logdate
Format: date
Width: 8
Allowable value: MM/DD/YY
Description: date when the task was performed

Field name: Logtime

Format: Character

Width: 5

Allowable value: HR:MN

Description: time when the task was performed

V. IMPLEMENTATION

A. INTRODUCTION

The introduction of a database as the central reservoir of data changes the organization's attitude with regard to data requirements and its management. This creates new skills to be learned and a need for a good implementation scheme. In this chapter, the issues of shipboard implementation and database administration are discuss.

B. SHIPBOARD IMPLEMENTATION

The implementation of a new system or a change in the old ways of doing things in an organization usually receives resistance and skepticism from the personnel involved; however, the security manager onboard a ship can easily overcome these problems due to the relatively small size of his department. The security manager and secret custodian must be aware of these implementation problems in order to correct problems as they arise.

Security consideration with regard to the TEMPEST requirements must be checked periodically. Although TEMPEST requirements do not apply to the application program in its present form, the security manager must keep abreast with any change in requirements. Quarterly training must be conducted in house for all users. A duplicate copy of database files (such as PUBS.DBF, PUBSINFO.DBF, and DESTRUCT.DBF) must be retrieved weekly for data accuracy and auditing. Program and backup disks must be stored in the safe for better security.

Finally, shipboard implemtation must be planned and monitored in order to make immediate changes when necessary. The users (sub-custodians) must be capable of learning the basic knowledge on how to use the proposed software program and the application program in this thesis.

1. Software Requirement for System Implementation

The dBASE III PLUS, which will be used for the implementation of the publication inventory system under discussion, is a relational DBMS for micro-computers. It contains its own extremely powerful programming language which permits the user to easily create his own application programs regardless of complexity.

2. Hardware Requirement for System Implementation

The system can operate on a 16-bit microcomputer that uses MS DOS, PC DOS version 2.0, or a newer version. For example, it could be an IBM PC, or an IBM PC/XT/AT or any other 16-bit microcomputer fully compatible with one of the above mentioned micro-computers.

256K of random access memory (RAM) is the minimum requirement of the supporting database package. The best configuration is one disk drive and one hard drive disk. A 10M (megabytes) hard disk is recommended because it is the most common, inexpensive, and reasonable configuration of a microcomputer. Although a hard disk is not required (because all files and application programs can be stored on a floppy disk based system), a hard disk will greatly improve the execution time of the programs used in this application. Any 80 column printer able to interface with the above mentioned microcomputers can be used.

C. DATABASE ADMINISTRATION

To facilitate effective use of the database, most organizations have staffed a position (and office) of database administration (DBA). This person is responsible for protecting the database while at the same time maximizing benefits to users [Ref. 4].

The secret custodian should be assigned as DBA. The DBA will be responsible for the accuracy, consistency, and timeliness of the data file. This function will require the secret custodian to provide database standards and enforce data activity policy. Therefore, the secret custodian must educate himself in order to effectively manage the database resources.

VI. SAMPLE PROGRAM PROCESSING

The proposed inventory management system is a menu driven program organized along functional lines. Each major function in the system corresponds to a section of this chapter and selection from the main menu. This chapter will incorporate the instruction on how to run the system; thus, it will serve as a substitute for the user's manual.

There is also a section in this chapter that discusses MAINPROG (or main program menu). It is important to read this section before starting to use the inventory management system since it describes in detail how to include the user's name in the access list of the program.

This manual is tailored to users with some knowledge in microcomputers. The steps to start the system are as follows:

a. Copy the two dBASE III PLUS diskettes the hard disk and insert the original third disk in the "A" drive.

b. At "C>" type "bdase" (no quotes needed) and then strike the return key. Once the system is loaded strike the return key once more. This will take you to the "assist" mode. You must get out of the assist mode by pressing the "esc" key. The "." (dot prompt) will appear on the screen. Type " set defa to a" (no quotes needed) and then hit the return key. Now you are ready to use the system.

c. At this point, type "do mainprog" followed by carriage return. A U.S. flag will appear momentarily then a brief greeting by the system. The system will prompt you for your own user's identification (to access the system see section MAINPROG of this chapter).

d. Follow the command messages which appear on the screen. The various options of the main function of the system are listed in the main menu. Select the number and follow the command message on the screen.

A. MAINPROG

The MAINPROG program has two functions: first, it will ask the user for his own access code or password. Once the user has succesfully logged into the system, a history of the user's session will be kept in the "USERLOG" file. To enter an authorized user:

a. Type "assist" at the "."(dot prompt) of bBASE III PLUS

b. Select the "Set Database file"

c. Select "a" drive, then select USERID.DBF

d. Select "Update", then select append

e. Enter the username and password

f. Save the new information (Ctrl-End)

g. Do step 1 to 3 and USERLOG.DBF

h) Do step 4. Enter username and save

Second, the main program will prompt the user with a straightforward menu driven selection process. The selection process does not require further information from the user.

The MAINPROG program presents the main menu on the screen(Figure 6.1). There are a total of six options. The user must refer to the respective section of this chapter for further information about the option in the menu.

```
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                    MAIN MENU                     :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                                                  :
:   1. HELP                                        :
:   2. UPDATE DATABASE                             :
:   3. REPORT GENERATOR                            :
:   4. ADHOC REQUEST                               :
:   5. END OF DATABASE                             :
:   6. EXIT TO DOS                                 :
:                                                  :
:     ENTER CHOICE (1-6) =>:                       :
:                                                  :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
```

Figure 6.1 Main Menu.

B. HELP

The help function allows the user to have on-line assitance while running the program. This help menu (Figure 6.2) will only be accessible from the main menu.

```
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                      HELP MENU                    :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                                                   :
:   1. UPDATE DATABASE                              :
:   2. REPORT GENERATOR                             :
:   3. ADHOC REQUEST                                :
:   4. EXIT TO MAIN MENU                            :
:                                                   :
:        ENTER CHOICE =>:                           :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
```

Figure 6.2 Help Menu.

C. UPDATE DATABASE

The update database function allows the user to add, edit, and delete any files in the menu provided in Figure 6.3. It is very important for the user to be exact with the information entered in the database because a mistake could have far-reaching implications to the validity of the data in the database.

```
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:        UPDATE  DATABASE  MENU                     :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                                                   :
: 1. INSERT RECORDS INTO PUBS FILE                  :
: 2. INSERT RECORDS INTO PUBSINFO FILE              :
: 3. INSERT RECORDS INTO DESTRUCT FILE              :
: 4. MODIFY RECORD FROM PUBS FILE                   :
: 5. MODIFY RECORD FROM PUBSINFO FILE               :
: 6. DELETE RECORDS PUBS & PUBSINFO FILES           :
: 7. EXIT TO MAIN MENU                              :
:                                                   :
:        ENTER CHOICE =>:                           :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
```

Figure 6.3 Update Database Menu

1. <u>Select Option "1" to Insert Pubs Record</u>

This allows the user to insert a new publication in the PUBS.DBF file. The user must be very careful when inserting the information into database. A control number must be provided by the user. The system will search for any duplication of control number in the database and will not allow the user to duplicate this number. Figure 6.4 shows the initial screen that will appear after the option 1 was selected.

```
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:         ADD NEW RECORD TO PUBS FILE         :
: WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW :
:                                             :
:                                             :
:       ENTER CONTROL NUMBER => 99999         :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
```

Figure 6.4 Query to Insert Pubs Record.

D. REPORT GENERATOR

The report function allows the user to obtain detail information about the inventory level of the all the publications. The user can access this section through the main menu. Figure 6.5 shows the "Database Report Generator Menu".

```
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                                             :
:       DATABASE REPORT GENERATOR MENU        :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                                             :
:   1.   ALLOWANCE LISTING OF PUBLICATION     :
:   2.   ALPHABETICAL LISTING OF PUBLICATION  :
:   3.   LIST OF PUBLICATIONS BY LOCATION     :
:   4.   LIST OF PUBLICATIONS BY CLASSIFICATION:
:   5.   LIST OF DESTRUCTION RECORDS          :
:   6.   EXIT TO MAIN MAIN MENU               :
:                                             :
:          ENTER CHOICE =>:                   :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
```

Figure 6.5 Database Report Generator Menu.

1. Select Option 1 to Get Allowance Listing

This section provides a list of publications sorted by document type with the corresponding allowance versus onboard listing. The output is shown below (Figure 6.6).

```
Page No.  1
09/24/87
                         USS BATTLESHIP REPORT
                         SORTED BY DOCUMENT TYPE
                         =====================

SHORT TITLE                        ALLOWANCE   ONBOARD
=========                          =========   =======

OPNAVINST 1120.1G                      1          1
SURFACINST 2210                        2          2
*** Total ***

                                       3          3
```

Figure 6.6 List of Publication Sorted by Document Type.

E. ADHOC REQUEST

The adhoc function allows the users to do special query on a specific publication. Although not all of the possible queries were provided, the user will be able to do basic inquiries from this menu. Figure 6.7 shows the available selection from the "Database Adhoc Menu".

```
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                                                 :
:              DATABASE ADHOC MENU                :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
:                                                 :
:   1. LOCATION OF SPECIFIC PUBLICATION           :
:   2. SUB-CUSTODIAN OF A PUBLICATION             :
:   3. ALLOWANCE OF SPECIFIC PUBLICATION          :
:   4. CLASSIFICATION OF SPECIFIC PUBLICATION     :
:   5. LAST UPDATE OF CERTAIN PUBLICATION         :
:                                                 :
:       ENTER CHOICE =>:                          :
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
```

Figure 6.7 Database Adhoc Menu.

VII. CONCLUSION

The purpose of this study was to develop a database system model suitable for implementation onboard Naval ships, and to aid in the inventory management of classified material in this environment.

The developed system must be considered a prototype model which may need further modification and extension in order to fit a specific user's need. Although the system must be considered a prototype in its present form, the application program will perform all the basic jobs of the sub-custodian with regard to inventory management.

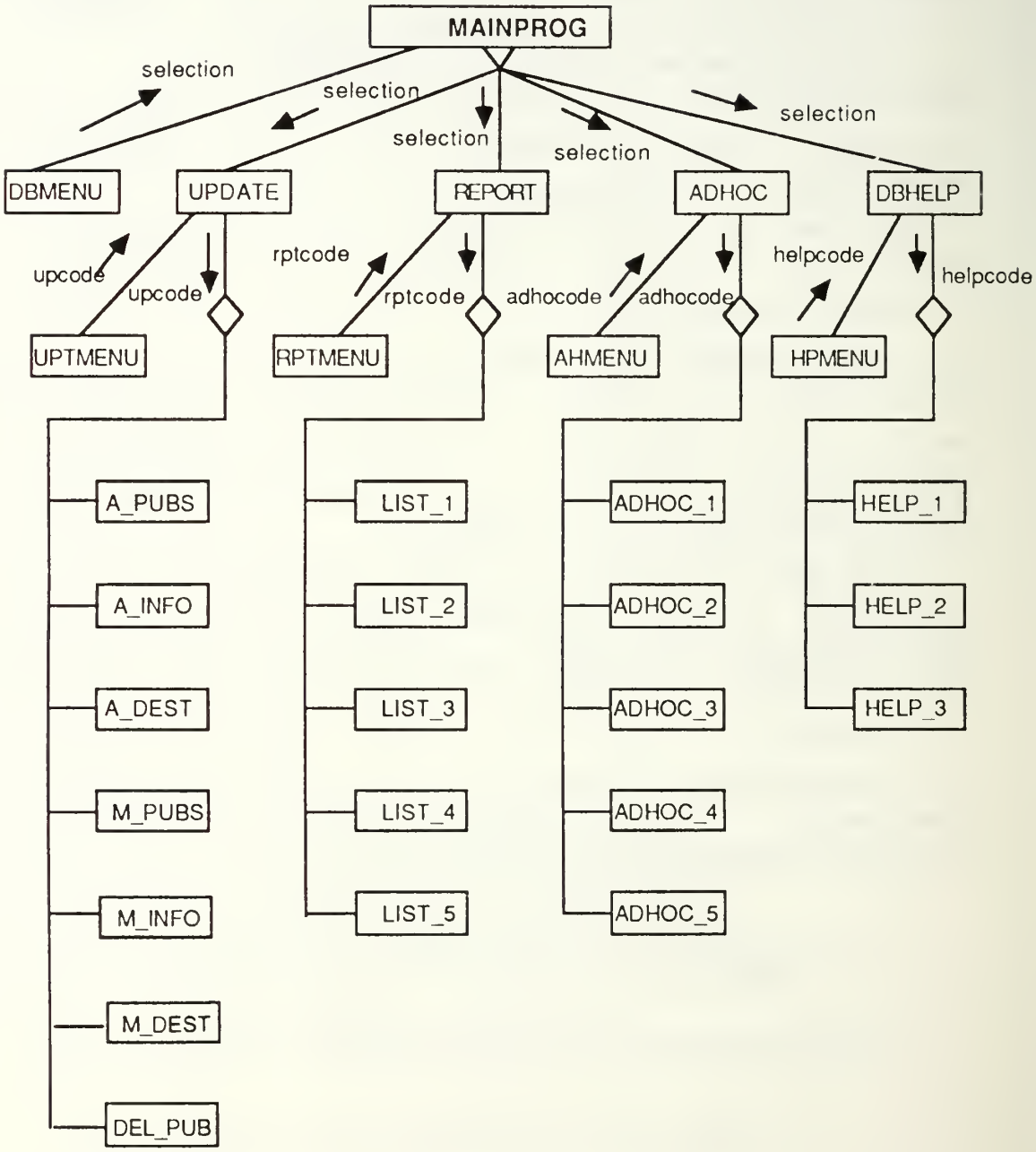
Since dBASE III PLUS was used in the application program of this thesis, a relational database must be used. A brief discussion of relational database model was reviewed in Chapter II, in order to give the user basic knowledge of the model.

The system is menu driven. Therefore, it is not only easy to use but also easy to modify, since both the user and programmer are directed to the desired point through the sequence of menus and sub-menus.

Finally, it is up to the security manager and secret custodian to monitor the effectiveness of this application program during the initial implementation stage and throughout its life cycle.

APPENDIX A

PROGRAM STRUCTURE CHART



APPENDIX B

PROGRAM

A. MAIN PROGRAM

```
********************   PROGRAM MAINPROG  ***************

* This is the main program, which controls the operation of
* the entire database system.

CLEAR

*  Initialization of the basic dBASE III functions

SET TALK OFF
SET BELL OFF
SET DELIMITER OFF
SET HEADING OFF
SET EXACT ON

* declare global variables

PUBLIC psw
STORE '        ' TO psw
@  5,18 SAY 'mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm'
@  6,18 SAY ': Security Manager dBASE Program :'
@  7,18 SAY 'mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm'
@  8,18 SAY ':                                 :'
@  9,18 SAY ':                                 :'
@ 10,18 SAY ': Please enter your               :'
@ 11,18 SAY ':                                 :'
@ 12,18 SAY ':                    $$$$$$$$     :'
@ 13,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'

* Check user's authorization

@ 12,28 SAY 'PASSWORD ->'
        SET CONSOLE OFF
        ACCEPT TO psw
        SET CONSOLE ON
USE userid
LOCATE FOR password = UPPER (psw)

* Unauthorized user. Exit to dBASE operating system.
```

```
IF  EOF ()
    SET COLOR TO W*
    @ 12,28 SAY '   UNAUTHORIZED USER '
    DO delay
    SET COLOR TO W
    QUIT
ENDIF

* Authorized user

STORE 'Y' TO main
DO flag
DO delay
DO headings

DO WHILE UPPER (main) = 'Y'
DO mmenu
    *  Perform appropriate function depending on user's choice.
    DO  CASE
        CASE selection = 1
            DO dbhelp
        CASE selection = 2
            DO update
        CASE selection = 3
            DO report
        CASE selection = 4
            DO adhoc
        CASE selection = 5
            STORE 'N' TO main
        CASE selection = 6
            CLEAR
            QUIT
   ENDCASE
ENDDO
SET TALK ON
SET DELIMITER ON
SET EXACT OFF
SET HEADING ON
SET BELL ON
CLEAR ALL
RETURN

* eof mainprog.prg
```

1. <u>Main Menu</u>

```
******************* PROGRAM MMENUM ********************

*  This program displays the system Main-Menu on the screen

CLEAR
SET TALK OFF
PUBLIC selection
STORE 0 TO selection

@   4,22 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwww'
@   5,22 SAY 'mm                      mm'
@   6,22 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwww'

SET COLOR TO W*
@   5,30 SAY 'MAIN MENU'
SET COLOR TO W

@   7,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@   8,18 SAY ':                                  :'
@   9,18 SAY ': 1. HELP/DOCUMENTATION            :'
@  10,18 SAY ': 2. UPDATE DATABASE               :'
@  11,18 SAY ': 3. REPORT GENERATOR              :'
@  12,18 SAY ': 4. ADHOC REQUESTS                :'
@  13,18 SAY ': 5. END OF DATABASE               :'
@  14,18 SAY ': 6. EXIT TO DOS                   :'
@  15,18 SAY ':                                  :'
@  16,18 SAY ':                                  :'
@  17,18 SAY ':                                  :'
@  18,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'

SET COLOR TO W+
@  17,20 SAY 'ENTER CHOICE (1-6) =>: 'GET selection;
PICT '9' RANGE 1,6
READ

SET COLOR TO W
SET TALK OFF
CLEAR
RETURN

*eof mmenu.prg
```

a. Update Sub-Menu

```
******************* PROGRAM UPDTMENU **********************
* This program display on the screen the update sub-menu

CLEAR
PUBLIC upcode
STORE 0 TO upcode
@  4,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  5,18 SAY ':                                                    :'
@  6,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
SET COLOR TO W*
@  5,30 SAY ' DATABASE UPDATE MENU'
SET COLOR TO W
@  7,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  8,18 SAY ':                                                    :'
@  9,18 SAY ':1. INSERT RECORDS INTO PUBS FILE                    :'
@ 10,18 SAY ':2. INSERT RECORDS INTO PUBSINFO FILE                :'
@ 11,18 SAY ':3. INSERT RECORDS INTO DESTRUCT FILE                :'
@ 12,18 SAY ':4. MODIFY RECORDS FROM PUBS FILE                    :'
@ 13,18 SAY ':5. MODIFY RECORDS FROM PUBSINFO FILE                :'
@ 14,18 SAY ':6. DELETE RECORDS FROM PUBS AND PUBSINFO FILES:'
@ 15,18 SAY ':7. EXIT TO MAIN MENU                                :'
@ 16,18 SAY ':                                                    :'
@ 17,18 SAY ':                                                    :'
@ 18,18 SAY ':                                                    :'
@ 19,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
SET COLOR TO W+
@ 17,30 SAY 'ENTER CHOICE =>:'GET upcode PICT '9' RANGE 1,7
        READ
SET COLOR TO W
CLEAR
RETURN
* eof updtmenu

******************* PROGRAM UPDATE ************************

CLEAR
SET TALK OFF
STORE '  ' TO updtcont
PUBLIC upcode
DO WHILE UPPER(updtcont) # 'N'
   DO updtmenu
   DO CASE
      CASE upcode = 1
           DO add_pubs
      CASE upcode = 2
           DO add_info
      CASE upcode = 3
           DO add_dest
      CASE upcode = 4
```

```
            DO mod_pubs
      CASE upcode = 5
            DO mod_info
      CASE upcode = 6
            DO del_pub
      CASE upcode = 7
            STORE 'N' TO updtcont
   ENDCASE
ENDDO
SET TALK ON
RETURN
*  eof update.prg
```

(1) Update Programs

```
****************** PROGRAM ADD_PUBS ************************

* This program insert records to PUBS file and records logged
* data into USERLOG file

CLEAR
SET TALK OFF
* open required files for processing
SELECT 1
USE PUBS index control
SELECT 2
USE USERID
SELECT 3
USE USERLOG

* dispaly a window on the screen
@  4,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  5,20 SAY ':                                              :'
@  6,20 SAY ': wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww  :'
@  7,20 SAY ':                                              :'
@  8,20 SAY ':                                              :'
@  9,20 SAY ':                                              :'
@ 10,20 SAY ':                                              :'
@ 11,20 SAY ':                                              :'
@ 12,20 SAY ':                                              :'
@ 13,20 SAY ':                                              :'
@ 14,20 SAY ':                                              :'
@ 15,20 SAY ':                                              :'
@ 16,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
STORE 'Y' TO add_cont
STORE .F. TO done
DO WHILE UPPER(add_cont) ='Y'
   @ 5,30 SAY 'ADD NEW RECORD TO PUBS FILE'

* initialize memory variables
   STORE '     '                        TO tctrl_no
   STORE '                          '   TO ts_title
   STORE ' '                            TO tlocker
   STORE '         '                    TO tclass
   STORE '        '                     TO tupdate
   STORE '            '                 TO tsub_cust

   @ 15,26 SAY 'ENTER CONTROL NUMBER =>'GET tctrl_no
   PICT '99999'
   READ
* check if record exist in PUBS file
SELECT 1
FIND &tctrl_no
```

```
IF EOF ()
   * record does not exits in the PUBS file
   * read user's inputs from the keyboard
   CLEAR
   @  2,00 TO 14,79 DOUBLE
   @  5,20 SAY 'ADDING NEW RECORD TO PUBS FILE '
   @  7,20 SAY ' CONTROL NUMBER :' GET tctrl_no
   @  8,20 SAY ' SHORT TITLE    :' GET ts_title
   @  9,20 SAY ' LOCKER NUMBER  :' GET tlocker
   @ 10,20 SAY ' CLASSIFICATION :' GET tclass
   @ 11,20 SAY ' SUB-CUSTODIAN  :' GET tsub_cust
   @ 12,20 SAY ' PUB LAST UPDATE:' GET tupdate PICT '99/99/99'
   READ

* append new record to PUBS file
APPEND BLANK
REPLACE ctrl_no WITH tctrl_no,s_title WITH ts_title
REPLACE locker WITH tlocker,class WITH tsub_cust
REPLACE update WITH CTOD(tupdate)
STORE .T. TO done

* control number exits in the PUBS file
ELSE
    SET COLOR TO W*
    @ 15,21 SAY '       CONTROL # ALREADY EXIST      '
    DO delay
    SET COLOR TO W
ENDIF
    SET COLOR TO W+
    @ 15,22 SAY '    MORE INSERTIONS? (Y/N)==>     '
    READ
    SET COLOR TO W
    SET CONSOLE OFF
    WAIT TO add_cont
    SET CONSOLE ON
ENDDO

* update USERLOG file
IF done
SELECT 2
LOCATE FOR password =UPPER(psw)
SELECT 3
APPEND BLANK
REPLACE username WITH B->username, task WITH 'INSERTION'
REPLACE progname WITH 'ADD_PUBS'
REPLACE logdate WITH DATE(),logtime WITH TIME()
ENDIF
CLEAR
CLOSE DATABASES
RETURN

* eof inserpub.prg
```

```
******************** PROGRAM ADD_INFO **********************

* This program inserts records to PUBSINFO file and records
* logged data into USERLOG file

CLEAR
SET TALK OFF
* open required files for processing
SELECT 1
USE PUBSINFO index ps_title
SELECT 2
USE USERID
SELECT 3
USE USERLOG

*  display a window on the screen
@   4,10 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@   5,10 SAY ':                                                   :'
@   6,10 SAY ':  wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww:'
@   7,10 SAY ':                                                   :'
@   8,10 SAY ':                                                   :'
@   9,10 SAY ':                                                   :'
@  10,10 SAY ':                                                   :'
@  11,10 SAY ':                                                   :'
@  12,10 SAY ':                                                   :'
@  13,10 SAY ':                                                   :'
@  14,10 SAY ':                                                   :'
@  15,10 SAY ':                                                   :'
@  16,10 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
?
?
STORE 'Y' TO addinfo
STORE .F. TO done
DO WHILE UPPER (addinfo) = 'Y'
   @ 5,18 SAY 'ADD NEW RECORD TO PUBSINFO FILE'

* initialize memory variables

  STORE '                         '            TO ts_title
  STORE '                                    ' TO tl_title
  STORE '  '                                   TO tallowance
  STORE '  '                                   TO tonboard
  STORE '       '                              TO tdoc_type
   @ 15,11 SAY '   SHORT TITLE ==>' GET ts_title
  READ

* check if record exist in PUBS file
SELECT 1
FIND &ts_title
IF EOF ()
   * record does not exist in PUBSINFO file
```

```
   * read user's inputs form keyboard
   CLEAR
   @   2,00 TO 14,79 DOUBLE
   @   4,18 SAY 'ADDING NEW RECORD TO PUBSINFO FILE '
   @   7,12 SAY 'SHORT TITLE:' GET ts_title
   @   8,12 SAY 'LONG TITLE: ' GET tl_title
   @   9,12 SAY 'ALLOWANCE:  ' GET tallowance
   @  10,12 SAY 'ONBOARD:    ' GET tonboard
   @  11,12 SAY 'DOC-TYPE:   ' GET tdoc_type
   READ

   * append new record to PUBSINFO file
   APPEND BLANK
   REPLACE s_title WITH ts_title, l_title WITH tl_title
   REPLACE allowance WITH VAL(tallowance)
   REPLACE onboard WITH VAL(tonboard)
   REPLACE doc_type WITH tdoc_type
   STORE .T. TO done
   * short title exist in the PUBSINFO file
ELSE
   @ 15,12 SAY '                                                '
   SET COLOR TO W*
   @ 15,12 SAY 'SHORT TITLE ALREADY EXIST'
   DO delay
   SET COLOR TO W

ENDIF
@ 15,12 SAY ' INSERT MORE RECORDS? (Y/N)=>'
SET CONSOLE OFF
WAIT TO addinfo
SET CONSOLE ON
ENDDO

* update USERLOG file
IF done
SELECT 2
LOCATE FOR password = UPPER(psw)
SELECT 3
APPEND BLANK
REPLACE username WITH B->username, task WITH 'INSERTIONS'
REPLACE progname WITH add_info, logdate WITH DATE()
REPLACE logtime WITH TIME()
ENDIF
CLEAR
CLOSE DATABASES
RETURN

* eof add_info
```

```
******************** PROGRAM ADD_DEST **********************

* This program inserts record to DESTRUCT file and update the
* USERLOG file

CLEAR
SET TALK OFF
* open required files for processing
SELECT 1
USE destruct INDEX dcontrol
SELECT 2
USE userid
SELECT 3
USE userlog

* display a window on the screen
@  8,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  9,20 SAY ':                                                      :'
@ 10,20 SAY ':   wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww   :'
@ 11,20 SAY ':                                                      :'
@ 12,20 SAY ':                                                      :'
@ 13,20 SAY ':                                                      :'
@ 14,20 SAY ':                                                      :'
@ 15,20 SAY ':                                                      :'
@ 16,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
STORE 'Y' TO add_cont
STORE .F. TO done
DO WHILE UPPER(add_cont) ='Y'
   @ 9,23 SAY 'ADD NEW RECORD TO DESTRUCT FILE'

   *  initialize memory variables
   STORE '     '                           TO tctrl_no
   STORE '                          '      TO ts_title
   STORE '        '                        TO td_date

   @ 15,25 SAY 'ENTER CONTROL # =>:'GET tctrl_no PICT '99999'
   READ
   SELECT 1
   FIND &tctrl_no
   IF EOF()
      * record does not exist in the DESTRUCT file
      * read user's input
      CLEAR
      @ 09,15 TO 15,60 DOUBLE
      @ 11,23 SAY 'CONTROL NUMBER:'GET tctrl_no
      @ 12,23 SAY 'SHORT TITLE   :'GET ts_title
      @ 13,23 SAY 'DATE DESTROYED:'GET td_date PICT '99/99/99'
      READ

      *append new record to DESTRUCT file
      APPEND BLANK
      REPLACE ctrl_no WITH tctrl_no, s_title WITH ts_title
```

```
      REPLACE d_destruct WITH CTOD(td_date)
      STORE .T. TO done
      *  control number exist in the DESTRUCT file
   ELSE
      SET COLOR TO W*
      @ 15,25 SAY ' RECORDS ALREADY EXIST'
      DO delay
      SET COLOR TO W
   ENDIF
   SET COLOR TO W+
   @  15,25 SAY '    MORE INSERTIONS? (Y/N)==>: '
   READ
   SET COLOR TO W
   SET CONSOLE OFF
   WAIT TO add_cont
   SET CONSOLE ON
ENDDO

* update USERLOG file
IF done
SELECT 2
LOCATE FOR PASSWORD = UPPER(psw)
SELECT 3
APPEND BLANK
REPLACE username WITH B->username, task WITH 'INSERTION'
REPLACE progname WITH 'INS_DEST', logdate WITH DATE()
REPLACE logtime WITH TIME ()
ENDIF
CLEAR
CLOSE DATABASES
RETURN
* eof add_dest.prg

******************* PROGRAM MOD_PUB ***********************

* This program modifies records in the PUBS file and updates
* USERLOG file

CLEAR
SET TALK OFF
* open required the files for processing
SELECT 1
USE pubs INDEX control
SELECT 2
USE userid
SELECT 3
USE userlog
STORE .F. TO done
STORE 'Y' TO modicont
SELECT 1
DO WHILE UPPER(modicont) = 'Y'
   STORE '      ' TO tctrl_no
```

```
*  display window on the screen
@   4,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@   5,20 SAY ':                                                :'
@   6,20 SAY ':    wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww   :'
@   7,20 SAY ':                                                :'
@   8,20 SAY ':                                                :'
@   9,20 SAY ':                                                :'
@  10,20 SAY ':                                                :'
@  11,20 SAY ':                                                :'
@  12,20 SAY ':                                                :'
@  13,20 SAY ':                                                :'
@  14,20 SAY ':                                                :'
@  15,20 SAY ':                                                :'
@  16,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@   5,26 SAY 'MODIFY RECORDS IN PUBS FILE'
@  15,26 SAY 'ENTER CONTROL NUMBER =>'GET tctrl_no PICT;
            '99999'
READ

* check if record to be modified exist in PUBS file
FIND &tctrl_no
IF .NOT. EOF ()
   * initialize memory variables
   STORE s_title  TO ts_title
   STORE locker   TO tlocker
   STORE class    TO tclass
   STORE update   TO tupdate
   STORE sub_cust TO tsub_cust

   * read user's input
   CLEAR
   @   2,05 TO 14,74 DOUBLE
   @   4,25 SAY 'MODIFYING PUBS FILE'
   @   5,25 SAY '*******************'
   @   7,20 SAY 'CONTROL NUMBER:' GET tctrl_no
   @   8,20 SAY 'SHORT TILTE   :' GET ts_title
   @   9,20 SAY 'LOCKER NUMBER :' GET tlocker
   @  10,20 SAY 'CLASSIFICATION:' GET tclass
   @  11,20 SAY 'LAST UPDATE   :' GET tupdate
   @  12,20 SAY 'SUB-CUSTODIAN :' GET tsub_cust
   READ

   * replace changed fields
   REPLACE s_title WITH ts_title, locker WITH tlocker
   REPLACE class WITH tclass, sub_cust WITH tsub_cust
   REPLACE update WITH tupdate
   STORE .T. TO done

   * record to be modified does not exist in PUBS file
   ELSE
       SET COLOR TO W*
```

```
        @ 15,25 SAY 'RECORD DOES NOT EXIST IN DATABASE'
        DO delay
        SET COLOR TO W
    ENDIF
    @ 15,25 SAY '                                  '
    @ 15,25 SAY 'MORE MODIFICATION? (Y/N)==>:'
    SET CONSOLE OFF
    WAIT TO modicont
    SET CONSOLE ON
    CLEAR
ENDDO

* update USERLOG file
IF done
   SELECT 2
   LOCATE FOR password = UPPER(psw)
   SELECT 3
   APPEND BLANK
   REPLACE username WITH B->username, task WITH 'MODIFICATION'
   REPLACE progname WITH 'MOD_PUB', logdate WITH DATE()
   REPLACE logtime WITH TIME()
ENDIF
SET TALK ON
CLEAR
CLOSE DATABASES
RETURN

* eof mod_pub.prg

******************* PROGRAM MOD_INFO.PRG *******************

* This program modifies records in the PUBSINFO file and
* updates USERLOG file

CLEAR
SET TALK OFF
* open required files for processing
SELECT 1
USE pubsinfo INDEX ps_title
SELECT 2
USE userid
SELECT 3
USE userlog
STORE .F. TO done
STORE 'Y' TO modicont
SELECT 1
DO WHILE UPPER(modicont) = 'Y'
   STORE '                          ' TO ts_title

* display a window on the screen
@  4,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  5,20 SAY ':                                              :'
```

```
@  6,20 SAY ': wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww :'
@  7,20 SAY ':                                                  :'
@  8,20 SAY ':                                                  :'
@  9,20 SAY ':                                                  :'
@ 10,20 SAY ':                                                  :'
@ 11,20 SAY ':                                                  :'
@ 12,20 SAY ':                                                  :'
@ 13,20 SAY ':                                                  :'
@ 14,20 SAY ':                                                  :'
@ 15,20 SAY ':                                                  :'
@ 16,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  5,25 SAY '   MODIFY RECORD IN PUBSINFO FILE '
@ 14,21 SAY ' ENTER SHORT TITLE =>:' GET ts_title
READ
@ 14,46 SAY '                        :          '
* check if record to be modified exists in PUBSINFO file
FIND &ts_title
IF .NOT. EOF()
   * record exists in PUBSINFO file
   * initialize memory variables
   STORE l_title   TO tl_title
   STORE allowance TO tallowance
   STORE onboard   TO tonboard
   STORE doc_type  TO tdoc_type

   * read user's data from keyboard
   CLEAR
   @  2,05 TO 13,74 DOUBLE
   @  4,25 SAY 'MODIFYING PUBSINFO FILE'
   @  5,25 SAY '***********************'
   @  7,10 SAY 'SHORT TITLE:' GET ts_title
   @  8,10 SAY 'LONG TITLE :' GET tl_title
   @  9,10 SAY 'ALLOWANCE  :' GET tallowance
   @ 10,10 SAY 'ONBOARD    :' GET tonboard
   @ 11,10 SAY 'DOC. TYPES :' GET tdoc_type
   READ
   * replace change fields
   REPLACE l_title WITH tl_title, allowance WITH tallowance
   REPLACE onboard WITH tonboard, doc_type WITH tdoc_type
   STORE .T. TO done

* record to be modified does not exist in the PUBSINFO file
ELSE
    SET COLOR TO W*
    @ 14,22 SAY 'RECORD DOES NOT EXIST IN THIS FILE   '
    DO delay
    SET COLOR TO W
ENDIF
@ 14,22 SAY '      MORE MODIFICATION? (Y/N)=>:        '
SET CONSOLE OFF
WAIT TO modicont
SET CONSOLE ON
```

```
CLEAR
ENDDO

* update USERLOG file
IF done
   SELECT 2
   LOCATE FOR password = UPPER(psw)
   SELECT 3
   APPEND BLANK
   REPLACE username WITH B->username, task WITH 'MODIFICATION'
   REPLACE progname WITH 'MOD_INFO', logdate WITH DATE()
   REPLACE logtime WITH TIME()
ENDIF
SET TALK ON
CLEAR
CLOSE DATABASES
RETURN

* eof mod_info.prg



 ******************** PROGRAM DEL_PUB ************************

* This program deletes records from the PUBS and PUBSINFO files
* and also update teh USERLOG file
CLEAR
SET TALK OFF
* open required files for proccessing
SELECT 1
USE pubs INDEX s_title
SELECT 2
USE pubsinfo INDEX ps_title
SELECT 3
USE userid
SELECT 4
USE userlog

* display a window on the screen
@  4,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  5,20 SAY '                                               :'
@  6,20 SAY ' wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww   :'
@  7,20 SAY '                                               :'
@  8,20 SAY '                                               :'
@  9,20 SAY '                                               :'
@ 10,20 SAY '                                               :'
@ 11,20 SAY '                                               :'
@ 12,20 SAY '                                               :'
@ 13,20 SAY '                                               :'
@ 14,20 SAY '                                               :'
@ 15,20 SAY '                                               :'
@ 16,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
```

```
STORE 'Y' TO deletecont
STORE 0 TO count1, count2
STORE .F. TO done
DO WHILE UPPER(deletecont) = 'Y'
   @ 5,22 SAY '                                          '
   @ 5,22 SAY 'DELETE RECORD FROM PUBS AND PUBSINFO FILE'
   STORE '                          ' TO ts_title
   @ 15,21 SAY 'ENTER SHORT TITLE =>:' GET ts_title
   READ
   *  check if the record to be deleted is in the PUBS file
   SELECT 1
   FIND &ts_title
   IF .NOT. EOF()
      * record exist in the PUBS file
      * display record to be deleted
      CLEAR
      @  2,05 TO 13,74 DOUBLE
      @  4,20 SAY ' DELETING RECORDS FROM THE PUBS FILE '
      @  5,20 SAY ' ********************************** '
      @  7,23 SAY 'CONTROL NUMBER :' GET ctrl_no
      @  8,23 SAY 'SHORT TITLE   :' GET s_title
      @  9,23 SAY 'LOCKER NUMBER :' GET locker
      @ 10,23 SAY 'CLASSIFICATION:' GET class
      @ 11,23 SAY 'LAST UPDATE   :' GET update
      @ 12,23 SAY 'SUB-CUSTODIAN :' GET sub_cust
      READ

      * user ensure it is the record to be deleted
      @ 15,40 SAY '                          '
      @ 15,21 SAY 'CONFIRM DELETIONS (Y/N)? ==>: '
      SET CONSOLE OFF
      WAIT TO confirm
      SET CONSOLE ON

      * delete the record from PUBS file
      IF UPPER(confirm) = 'Y'
         DELETE
         STORE .T. TO done
         STORE count1 + 1 TO count1

         * delete the record from PUBSINFO file
         SELECT 2
         FIND &ts_title
         IF .NOT. EOF()
            DELETE
            STORE count2 +1 count2
         ENDIF

      * do not delete record from PUBS file
      ELSE
         STORE 'F' TO deletecont
```

```
         CLEAR
      ENDIF

   *  record to be deleted does not exist in database
   ELSE
      @ 15,40 SAY '                                  '
      @ 15,21 SAY 'RECORD DOES NOT EXIST IN DATABASE'
      DO delay
      @ 15,25 SAY '                                  '
   ENDIF
   @ 15,21 SAY '    MORE DELETIONS? (Y/N) =>:'
   SET CONSOLE OFF
   WAIT TO deletecont
   SET CONSOLE ON
   CLEAR
ENDDO

* pack database files for permanent deletion of files deleted
* previously by the program

SELECT 1
IF count1 # 0
   @ 12,20 SAY ' DATABASE FILES ARE IN THE PROCESS OF BEING'
   @ 13,20 SAY ' PACKED AND REINDEXED. '
   @ 15,20 SAY ' Please be patient. It may take a while'
   PACK
   SELECT 2
   IF count2 # 0
      PACK
   ENDIF
ENDIF
CLOSE DATABASES


* update USERLOG
IF done
   SELECT 3
   USE userid
   LOCATE FOR PASSWORD = UPPER(psw)
   SELECT 4
   USE userlog
   APPEND BLANK
   REPLACE username WITH C->username, task WITH 'DELETION'
   REPLACE progname WITH 'DEL_PUB',logdate WITH DATE ()
   REPLACE logtime WITH time()
ENDIF
SET TALK ON
CLEAR
CLOSE DATABASES
RETURN

* eof del_pub.prg
```

b. Report Sub-Menu

```
******************** PROGRAM RPTMENU ************************

*  This program displays on the screen the database report
*  generator sub-menu.

CLEAR
PUBLIC rptcode
STORE 0 TO rptcode
@  2,14 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  3,14 SAY 'mm                                          mm'
@  4,14 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
SET COLOR TO W*
@  3,20 SAY 'DATABASE REPORT GENERATOR MENU'
SET COLOR TO W
@  6,14 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  7,14 SAY ':                                          :'
@  8,14 SAY ': 1. ALLOWANCE LISTING OF PUBLICATIONS     :'
@  9,14 SAY ': 2. ALPHABETICAL LISTING OF PUBLICATIONS  :'
@ 10,14 SAY ': 3. LIST OF PUBLICATIONS BY LOCATIONS     :'
@ 11,14 SAY ': 4. LIST OF PUBLICATIONS BY SUB-CUSTODIAN :'
@ 12,14 SAY ': 5. LIST OF DESTRUCTION RECORDS           :'
@ 13,14 SAY ': 6. EXIT TO MAIN MENU                     :'
@ 14,14 SAY ':                                          :'
@ 15,14 SAY ':                                          :'
@ 16,14 SAY ':                                          :'
@ 17,14 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
SET COLOR TO W+
@ 16,16 SAY 'ENTER CHOICE =>:' GET rptcode PICT '9' RANGE 1,6
        READ
SET COLOR TO W
RETURN

* eof rptmenu.prg


******************** PROGRAM REPORT **********************

CLEAR
STORE '  'TO rptcont
PUBLIC rptcode
DO WHILE rptcont # 'n'
   DO rptmenu
   DO CASE
      CASE rptcode = 1
           DO list_1
      CASE rptcode = 2
           DO list_2
      CASE rptcode = 3
           DO list_3
```

```
      CASE rptcode = 4
           DO list_4
      CASE rptcode = 5
           DO list_5
      CASE rptcode = 6
           STORE 'n' TO rptcont
    ENDCASE
ENDDO
RETURN

* eof report.prg
```

(1) LISTING/REPORT PROGRAMS

```
******************* PROGRAM LIST_1 ************************

* This program provides screen display or printer output of the
* total number of publications that are on the allowance list
* against what is onboard by document type.

CLEAR
SET TALK OFF
STORE .F. TO printer
STORE ' ' TO done
STORE ' ' TO ans
USE pubsinfo INDEX doc_type
DO WHILE UPPER(done) # 'N'
   @  3,10 TO 9,65 DOUBLE
   @  4,20 SAY ' DO YOU WANT TO PRINT OUTPUT ? '
   @  5,20 SAY ' PRINTER = P                   '
   @  6,20 SAY ' SCREEN = S                    '
   @  8,25 SAY ' ENTER S OR P =>:' GET ans
   READ
   IF UPPER(ans) = 'P'
      STORE .T. TO printer
      CLEAR
      @ 15,20 SAY 'TURN YOUR PRINTER ON'
      WAIT TO cont
   ENDIF

   STORE ' ' TO choice
   CLEAR
   @  7,10 TO 16,65 DOUBLE
   @  9,21 SAY '   SELECT THE DOCUMENT TYPE  '
   @ 11,21 SAY '   Secret = S                '
   @ 12,21 SAY '   Intelligence = I          '
   @ 13,21 SAY '   NTP/ATP Pubs = P          '
   @ 15,25 SAY 'ENTER CHOICE ==>:' GET choice
   READ
```

```
   IF printer
       DO CASE
          CASE UPPER(choice) = 'S'
               SET PRINTER ON
               REPORT FORM list1 FOR doc_type = "SECRET"
               SET PRINTER OFF
          CASE UPPER(choice) = 'I'
               SET PRINTER ON
               REPORT FORM list1 FOR doc_type = "INTEL"
               SET PRINTER OFF
          CASE UPPER(choice) = 'P'
               SET PRINTER ON
               REPORT FORM list1 FOR doc_type = "NWP"
               SET PRINTER OFF
       ENDCASE
   ELSE
       DO CASE
          CASE UPPER(choice) = 'S'
                REPORT FORM list1 FOR doc_type = "SECRET"
                ?
                WAIT TO cont
          CASE UPPER(choice) = 'I'
                REPORT FORM list1 FOR doc_type = "INTEL"
                ?
                WAIT TO cont
          CASE UPPER(choice) = 'P'
                REPORT FORM list1 FOR doc_type = "NWP"
                ?
                WAIT TO cont
       ENDCASE
   ENDIF
   GO TOP
   CLEAR
   STORE ' ' TO all_done
   @ 15,20 SAY ' DO YOU WANT TO SEE OR PRINT '
   @ 16,20 SAY ' THE LIST AGAIN ?            '
   @ 18,20 SAY ' ENTER "Y" OR "N" =>:' GET all_done
   READ
   IF UPPER(all_done) # 'Y'
      STORE 'N' TO done
   ENDIF
ENDDO

CLEAR
CLOSE DATABASES
RETURN

* eof list_1.prg


****************** PROGRAM LIST_2 ************************
```

```
* This program provides display or printer output of the
* publications in an alphabetical order. Short title and
* long title

CLEAR
SET TALK OFF
STORE .F. TO printer
STORE ' ' TO done
STORE ' ' TO ans
USE pubsinfo INDEX ps_title
DO WHILE UPPER(done) # 'N'
   @  3,10 TO 9,65 DOUBLE
   @  4,20 SAY ' DO YOU WANT TO PRINT OUTPUT ? '
   @  5,20 SAY ' PRINTER = P                    '
   @  6,20 SAY ' SCREEN = S                     '
   @  8,25 SAY ' ENTER S OR P =>:' GET ans
   READ
   IF UPPER(ans) = 'P'
      STORE .T. TO printer
      CLEAR
      @ 15,20 SAY 'TURN YOUR PRINTER ON'
      WAIT TO cont
   ENDIF
   IF printer
      SET PRINTER ON
      REPORT FORM list2
      SET PRINTER OFF
   ELSE
      CLEAR
      REPORT FORM list2
      WAIT TO cont
   ENDIF
   GO TOP
   CLEAR
   STORE ' ' TO all_done
   @ 15,20 SAY ' DO YOU WANT TO SEE OR PRINT '
   @ 16,20 SAY ' THE LIST AGAIN '
   @ 18,20 SAY ' ENTER "Y" OR "N" =>:' GET all_done
   READ
   IF UPPER(all_done) # 'Y'
      STORE 'N' TO done
   ENDIF
ENDDO
CLEAR
CLOSE DATABASES
RETURN

* eof list_2.prg


******************** PROGRAM LIST_3 ********************
CLEAR
```

```
SET TALK OFF
STORE .F. TO printer
STORE ' ' TO done
STORE ' ' TO ans
USE pubs INDEX locker
DO WHILE UPPER(done) # 'N'
   @  3,10 TO 9,65 DOUBLE
   @  4,20 SAY ' DO YOU WANT TO PRINT OUTPUT ? '
   @  5,20 SAY ' PRINTER = P                   '
   @  6,20 SAY ' SCREEN = S                    '
   @  8,25 SAY ' ENTER S OR P =>: ' GET ans
   READ
   IF UPPER(ans) = 'P'
      STORE .T. TO printer
      CLEAR
      @  15,20 SAY ' TURN YOUR PRINTER ON '
      WAIT TO cont
   ENDIF
   IF printer
      SET PRINTER ON
      REPORT FORM list3
      SET PRINTER OFF
   ELSE
      CLEAR
      REPORT FORM list3
      WAIT TO cont
   ENDIF
   GO TOP
   CLEAR
   STORE ' ' TO all_done
   @ 15,20 SAY ' DO YOU WANT TO SEE OR PRINT '
   @ 16,20 SAY ' THE LIST AGAIN '
   @ 18,20 SAY ' ENTER "Y" OR "N" =>:' GET all_done
   READ
   IF UPPER(all_done) # 'Y'
      STORE 'N' TO done
   ENDIF
ENDDO
CLEAR
CLOSE DATABASES
RETURN

* eof list_3.prg


********************** PROGRAM LIST_4 ****************
* This program provides display or printed output of the
* publications by classification

CLEAR
SET TALK OFF
```

```
STORE .F. TO printer
STORE ' ' TO done
STORE ' ' TO ans
USE pubs INDEX class
DO WHILE UPPER(done) # 'N'
   CLEAR
   @  3,10 TO 9,65 DOUBLE
   @  4,20 SAY ' DO YOU WANT TO PRINT OUTPUT ? '
   @  5,20 SAY ' PRINTER = P                   '
   @  6,20 SAY ' SCREEN = S                    '
   @  8,25 SAY ' ENTER S OR P =>: ' GET ans
   READ
   IF UPPER(ans) = 'P'
      STORE .T. TO printer
      CLEAR
      @ 15,20 SAY ' TURN YOUR PRINTER ON '
      WAIT TO cont
   ENDIF
   IF printer
      SET PRINT ON
      REPORT FORM list4
      SET PRINTER OFF
   ELSE
      CLEAR
      REPORT FORM list4
      WAIT TO cont
   ENDIF
   GO TOP
   CLEAR
   STORE ' ' TO all_done
   @  15,20 SAY ' DO YOU WANT TO SEE OR PRINT '
   @  16,20 SAY ' THE LIST AGAIN '
   @  18,20 SAY ' ENTER "Y" OR "N" =>:' GET all_done
   READ
   IF UPPER(all_done) # 'Y'
      STORE 'N' TO done
   ENDIF
ENDDO
CLEAR
CLOSE DATABASES
RETURN

* eof list_4.prg


****************** PROGRAM LIST_5 ************************
* This program provides display or printer output of the
* destruction record file

CLEAR
SET TALK OFF
STORE .F. TO printer
```

```
STORE ' ' TO done
STORE ' ' TO ans
USE destruct INDEX date
DO WHILE UPPER(done) # 'N'
   CLEAR
   @  3,10 TO 9,65 DOUBLE
   @  4,20 SAY ' DO YOU WANT TO PRINT OUTPUT ? '
   @  5,20 SAY ' PRINTER = P                   '
   @  6,20 SAY ' SCREEN = S                    '
   @  8,25 SAY ' ENTER S OR P =>:' GET ans
   READ
   IF UPPER(ans) = 'P'
      STORE .T. TO printer
      CLEAR
      @  15,20 SAY ' TURN YOUR PRINTER ON '
      WAIT TO cont
   ENDIF
   IF printer
      SET PRINT ON
      REPORT FORM list5
      SET PRINTER OFF
   ELSE
      CLEAR
      REPORT FORM list5
      WAIT TO cont
   ENDIF
   GO TOP
   CLEAR
   STORE ' ' TO all_done
   @ 15,20 SAY ' DO YOU WANT TO SEE OR PRINT '
   @ 16,20 SAY ' THE LIST AGAIN '
   @ 18,20 SAY ' ENTER "Y" OR "N" =>:' GET all_done
   READ
   IF UPPER(all_done) # 'Y'
      STORE 'N' TO done
   ENDIF
ENDDO
CLEAR
CLOSE DATABASES
RETURN

* eof list_5.prg
```

c. Adhoc Sub-Menu

```
******************** PROGRAM HOCMENU ************************
* This program display on the screen the adhoc sub-menu

CLEAR
SET TALK OFF
PUBLIC adhocode
STORE 0 TO adhocode
@  5,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  6,18 SAY ':                                                    :'
@  7,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
SET COLOR TO W*
@  6,30 SAY ' DATABSE ADHOC MENU '
SET COLOR TO W
@  8,18 SAY 'mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm'
@  9,18 SAY ':                                                    :'
@ 10,18 SAY ': 1. LOCATION OF SPECIFIC PUBLICATION                :'
@ 11,18 SAY ': 2. SUB-CUSTODIAN OF A PUBLICATION                  :'
@ 12,18 SAY ': 3. ALLOWANCE OF SPECIFIC PUBLICATION               :'
@ 13,18 SAY ': 4. CLASSIFICATION OF SPECIFIC PUBLICATION          :'
@ 14,18 SAY ': 5. LAST UPDATE OF CERTAIN PUBLICATION              :'
@ 15,18 SAY ': 6. EXIT TO MAIN MENU                               :'
@ 16,18 SAY ':                                                    :'
@ 17,18 SAY ':                                                    :'
@ 18,18 SAY ':                                                    :'
@ 19,18 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
SET COLOR TO W+
@ 17,30 SAY 'ENTER CHOICE =>:' GET adhocode PICT '9' RANGE 1,7
READ
SET COLOR TO W
CLEAR
SET TALK ON
RETURN
* eof hocmenu.prg


******************** PROGRAM ADHOC **************************
* This program call the proper adhoc program selected from the
* adhoc sub-menu by users

CLEAR
STORE '  ' TO adhcont
PUBLIC adhocode
DO WHILE adhcont # 'n'
   DO hocmenu
   DO CASE
      CASE adhocode = 1
           DO adhoc_1
      CASE adhocode = 2
           DO adhoc_2
```

```
      CASE adhocode = 3
           DO adhoc_3
      CASE adhocode = 4
           DO adhoc_4
      CASE adhocode = 5
           DO adhoc_5
      CASE adhocode = 6
           STORE 'n' TO adhcont
   ENDCASE
ENDDO
RETURN

* eof adhoc.prg
```

(1) ADHOC PROGRAMS

```
****************** PROGRAM ADHOC_1 **********************
* To accept a specified publication and provide their locker
* location

CLEAR
SET TALK OFF
@ 4,10 TO 13,69 DOUBLE

@  6,18 SAY '   The purpose of this adhoc function  # 1 is'
@  7,18 SAY 'to locate a specific publication for the user.'
@  8,18 SAY 'The user must provide the exact short title  '
@  9,18 SAY 'of the publication as previously enter in the'
@ 10,18 SAY 'database. Otherwise the search will be un-  '
@ 11,18 SAY 'successful and frustrating. '
?
?
WAIT TO cont
CLEAR

 * open required files for processing
SELECT 1
USE pubs INDEX s_title
SELECT 2
USE userid
SELECT 3
USE userlog
STORE .F. TO done
STORE 'Y' TO adhoc1
SELECT 1
DO WHILE UPPER(adhoc1) ='Y'
   ACCEPT 'ENTER SHORT TITLE :   ' TO s_title
   FIND &s_title
   IF .NOT. EOF ()
     CLEAR
     @ 7,05 SAY ' SHORT TITLE '
     @ 7,30 SAY ' LOCKER # '
     DISPLAY s_title, locker
     DO delay
   ELSE
     SET COLOR TO W*
     @ 15,25 SAY ' NO SUCH SHORT TITLE '
     DO delay
     SET COLOR TO W
   ENDIF
   @ 15,25 SAY ' MORE INQUIRY?  (Y/N)==>:' GET adhoc1
   READ
   SET CONSOLE OFF
   WAIT TO achoc1
   SET CONSOLE ON
   CLEAR
```

```
ENDDO
CLEAR
SET TALK ON
RETURN

* eof adhoc_1.prg


******************* PROGRAM ADHOC_2.PRG *******************
* This program accepts short title of a specific publication
* and provide the user the name of the current sub-custodian

CLEAR
SET TALK OFF
@  4,10 TO 13,69 DOUBLE
@  6,18 SAY 'The purpose of this ADHOC function # 2 is to'
@  7,18 SAY 'locate the current sub-custodian of a specific'
@  8,18 SAY 'publication. The user must provide the exact '
@  9,18 SAY 'short title of the publication as previously'
@ 10,18 SAY 'enter in this database. Otherwise the search'
@ 11,18 SAY 'will be unsuccesful and frustrating.'
?
?
WAIT TO cont
CLEAR

STORE .F. TO done
STORE 'Y' TO adhoc2
DO WHILE UPPER(adhoc2) = 'Y'
   ACCEPT 'ENTER SHORT TITLE: ' TO s_title
   USE pubs INDEX s_title
   FIND &s_title
   IF .NOT. EOF ()
      DISPLAY s_title, sub_cust
      DO delay
   ELSE
     SET COLOR TO W*
     @ 15,25 SAY ' NO SUCH SHORT TITLE '
     DO delay
     SET COLOR TO W
   ENDIF
   @ 15,25 SAY '  MORE INQUIRY?  (Y/N)==>:' GET adhoc2
   READ
   SET CONSOLE OFF
   WAIT TO adhoc2
   SET CONSOLE ON
   CLEAR
ENDDO
CLEAR
SET TALK ON
RETURN
```

```
* eof adhoc_2.prg


******************* PROGRAM ADHOC_3 **********************
* This program accepts short title of a specific publication
* and provide the user the current allowance and current
* onboard inventory.

CLEAR
SET TALK OFF
TEXT
     The purpose of this ADHOC function # 3 is to provide
     current allowance vs current inventory of a specified
     publication. The user must enter the exact short
     title of the publication.
ENDTEXT
WAIT TO cont
CLEAR

STORE .F. TO done
STORE 'Y' TO adhoc
DO WHILE UPPER(adhoc) = 'Y'
   ACCEPT ' ENTER SHORT TITLE: ' TO s_title
   USE pubsinfo INDEX s_title
   FIND &s_title
   IF .NOT. EOF ()
      DISPLAY s_title, allowance, onboard
      DO delay
   ELSE
      SET COLOR TO W*
      @ 15,25 SAY ' NO SUCH SHORT TITLE '
      DO delaY
      SET COLOR TO W
   ENDIF
   @ 15,25 SAY ' MORE INQUIRY? (Y/N) ==>:' GET adhoc
   READ
   SET CONSOLE OFF
   WAIT TO adhoc
   SET CONSOLE ON
   CLEAR
ENDDO
CLEAR
SET TALK ON
RETURN

* eof adhoc_3.prg


******************* PROGRAM ADHOC_4.PRG ******************
* This program accepts short title of a specific publication
* and provide the user the classification category in return
```

```
CLEAR
SET TALK OFF
TEXT
     The purpose of this ADHOC function # 4 is to give the
     classification category of a specific publication. The
     user must provide the exact short title of the publication
     as previously enter in this database. Otherwise the search
     will be unsuccessful.
ENDTEXT
WAIT TO cont
CLEAR

STORE .T. TO done
STORE 'Y' TO adhoc
DO WHILE UPPER(adhoc) = 'Y'
   ACCEPT 'ENTER SHORT TITLE: ' TO s_title
   USE pubs INDEX s_title
   FIND &s_title
   IF .NOT. EOF ()
      DISPLAY s_title, class
      DO delay
   ELSE
      SET COLOR TO W*
      @ 15,25 SAY ' NO SUCH SHORT TITLE '
      DO delay
      SET COLOR TO W
   ENDIF
   @ 15,25 SAY '  MORE INQUIRY?   (Y/N)==>:'GET adhoc
   READ
   SET CONSOLE OFF
   WAIT TO adhoc
   SET CONSOLE ON
   CLEAR
ENDDO
CLEAR
SET TALK ON
RETURN

* eof adhoc_4.prg


******************* PROGRAM ADHOC_5 ************************
* This program accepts short title of a specific publication
* and provide the last update made on this publication

CLEAR
SET TALK OFF
TEXT
     The purpose of this ADHOC function # 5 is to provide the
     date of last update made on the specified publication.
     The user must enter the exact short title of the publi-
     cation as previously enter in this database. Otherwise,
```

```
     the search will be unsuccessful.
ENDTEXT
WAIT TO cont
CLEAR

STORE .F. TO done
STORE 'Y' TO adhoc
DO WHILE UPPER(adhoc) = 'Y'
   ACCEPT ' ENTER SHORT TITLE: ' TO s_title
   USE pubs INDEX s_title
   FIND &s_title
   IF .NOT. EOF()
      DISPLAY s_title, update
      DO delay
   ELSE
      SET COLOR TO W*
      @ 15,25 SAY 'NO SUCH SHORT TITLE'
      DO delay
      SET COLOR TO W
   ENDIF
   @ 15,25 SAY '  MORE INQUIRY?  (Y/N)==>:' GET adhoc
   READ
   SET CONSOLE OFF
   WAIT TO adhoc
   SET CONSOLE ON
   CLEAR
ENDDO
CLEAR
SET TALK ON
RETURN
* eof adhoc_5.prg
```

d. Help Sub-Menu

```
******************** PROGRAM HPMENU *************************
* This program display on the screen the help sub-menu
* called by DBHELP.PRG

CLEAR
SET TALK OFF
PUBLIC HELPCODE
STORE 0 TO HELPCODE
@  6,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
@  7,20 SAY ':                                                :'
@  8,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'
SET COLOR TO W*
@  7,30 SAY ' DATABASE HELP MENU '
SET COLOR TO W
@  9,20 SAY 'mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm'
@ 10,20 SAY ':                                                :'
@ 11,20 SAY ': 1. UPDATE HELP                                 :'
@ 12,20 SAY ': 2. REPORT HELP                                 :'
@ 13,20 SAY ': 3. ADHOC HELP                                  :'
@ 14,20 SAY ': 4. EXIT TO MAIN MENU                           :'
@ 15,20 SAY ':                                                :'
@ 16,20 SAY ':                                                :'
@ 17,20 SAY ':                                                :'
@ 18,20 SAY 'wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww'

SET COLOR TO W+
@ 17,30 SAY 'ENTER CHIOCE =>:' GET helpcode PICT '9' RANGE 1,4
READ
SET COLOR TO W
SET TALK ON
CLEAR
RETURN

* eof hpmenu.prg


************************* PROGRAM DBHELP *******************
*  Provide an online help reference.
*  called by: mmenu.prg

CLEAR
SET TALK OFF

@  2,00 TO 11,79 DOUBLE

@  4,20 SAY '   HH   HH    EEEEEEE  LL         PPPPPP  '
@  5,20 SAY '   HH   HH    EE       LL         PP   PP '
@  6,20 SAY '   HHHHHHH    EEEEEE   LL         PpPPPP  '
@  7,20 SAY '   HH   HH    EE       LL         PR      '
```

```
@  8,20 SAY '   HH   HH    EEEEEEE  LLLLLLLL PP       '
@  9,20 SAY '   ***********************************   '
?
?
DO delay

TEXT
   The security manager inventory program will provide
   information of command's classified publications. To
   see specific information and guidance on each type of
   query, type the appropriate choice or task code number.
   The following information pertains to many of the reports
   and summaries:

   Attribute   type   # chars.  Example
   CTRL_NO      N        4      192
   S_TITLE      C       17      NTP 4(G)
   L_TITLE      C       25      NAVAL COMMUNICATIONS PROC
   LOCKER       N        1      3
   CLASS        C        6      UNCLAS, CONF, SECRET
   UPDATE       D        -      05/20/86
   SUB_CUST     C       10      PO2 SMITH, SCPO JONES
   ALLOWANCE    N        1      3
   ONBOARD      N        1      3
   DOC_TYPE     C        7      NTP/ATP, INTEL, SECRET

ENDTEXT
?
?
WAIT TO cont
STORE '  ' TO helpcont
PUBLIC helpcode
DO WHILE helpcont # 'n'
   DO hpmenu
   DO CASE
      CASE helpcode = 1
           DO help_1
      CASE helpcode = 2
           DO help_2
      CASE helpcode = 3
           DO help_3
      CASE helpcode = 4
           STORE 'n' TO helpcont
    ENDCASE
ENDDO
CLEAR
RETURN

* eof dbhelp.prg
```

(1) HELP PROGRAMS

```
******************* PROGRAM HELP_1 ************************
* This program will provide an on-line help note for user
* with regards to updating the database.

CLEAR
SET TALK OFF
TEXT


    The "Update Database Menu" is divided into 6 edit choices.
    Option 1 will allow the user to enter the new publication
    into the PUBS.INFO file. You must provide the exact infor-
    mation to the database. Failure to do so will affect the
    effectiveness of the this system. The rest of the options
    in this menu will be very similar to option 1 and will be
    simple if you just follow the instruction provided in the
    program. Wait for the prompt and enter the data with care.


ENDTEXT

WAIT TO cont
CLEAR
SET TALK OFF
RETURN

* eof help_1.prg


******************* PROGRAM HELP_2 ************************
*  This program will provide on-line help note with regards
* to the Report generator menu

CLEAR
SET TALK OFF
TEXT


    The "Report Generator Menu" provide five listing for the
    the user of this system. The first listing is the "Report
    listing of publication sorted by document type and total
    by allowance vs onboard. The second listing is the "Alpha-
    betical Listing of all Publication". The third listing is
    the "Listing of of Publication Sorted by Locker Number".
    The fourth listing is the list of "Publication Sorted by
    Classification Types". Finally the "Destruction Record
    Sorted by Date". All these listings are formated into a
    report format (i.e., LIST_1.FRM) which could be modified
    easily with the assist query of the dBASE III.
```

```
ENDTEXT

WAIT TO cont
CLEAR
RETURN

* eof help_2.prg


******************* PROGRAM HELP_3 ************************
* This program will provide an on-line help note with regards
* to database adhoc menu.

CLEAR
SET TALK OFF
TEXT


    The "Database adhoc menu" is divided into 5 adhoc queries.
    After making the selection the user will prompt to answer
    question ask by the system. The adhoc programs were prov-
    ided to the users as sample queries that could valuable to
    the user.


ENDTEXT

WAIT TO cont
CLEAR
RETURN

* eof help_3.prg
```

B. MISCELLANEOUS PROGRAMS

```
******************* PROGRAM FLAG *************************

* This program displays the U.S. flag for the main program

CLEAR

@  1,10 SAY '_____________________________________________'
@  2,10 SAY '*   *   *   *   *   * '
@  3,10 SAY '  *   *   *   *   *  '
@  4,10 SAY '*   *   *   *   *   * '
@  5,10 SAY '  *   *   *   *   *  '
@  6,10 SAY '*   *   *   *   *   * '
@  7,10 SAY '  *   *   *   *   *  '
@  8,10 SAY '*   *   *   *   *   * '
@  9,10 SAY '  *   *   *   *   *  '
@ 10,10 SAY '*   *   *   *   *   * '
@ 16,10 SAY '_____________________________________________'

SET COLOR TO W+
STORE '                                  ' TO blank1
STORE '                        ' TO blank2
@  3,28 GET blank1
@  5,28 GET blank1
@  7,28 GET blank1
@  9,28 GET blank1
@ 11,10 GET blank2
@ 13,10 GET blank2
@ 15,10 GET blank2
@ 11,28 GET blank1
@ 13,28 GET blank1
@ 15,28 GET blank1

SET COLOR TO W
@ 20,18 SAY ' UNITED STATES NAVY SHIP '
RETURN

* eof flag.prg


****************** PROGRAM HEADING **********************
* This program displays on the screen the program information
* headings

SET COLOR TO W+
CLEAR

@  2,00 TO 18,79 DOUBLE
```

```
@  3,20 SAY ' ********************************** '
@  4,20 SAY ' SECURITY MANAGER DATABASE SYSTEM '
@  5,20 SAY ' ********************************** '
@  6,15 SAY '   The Security Manager Inventory System is'
@  7,13 SAY ' an application program design to support the'
@  8,13 SAY ' following activities:'
@ 10,13 SAY ' 1. Provide publications listings by control,'
@ 11,13 SAY ' number short title, location, last update,'
@ 12,13 SAY ' classification and alphabetic listing.'
@ 13,13 SAY ' 2. Provide information on specific publication.'
@ 14,13 SAY ' 3. Provide total no. of publications by locker.'
@ 15,13 SAY ' 4. Provide destruction records of publications.'
@ 16,13 SAY ' 5. Answer Ad Hoc queries by the user.'
?
?
WAIT TO continue
CLEAR
SET COLOR TO W
RETURN

* eof headings.prg


******************** PROGRAM DELAY ************************
* This program provides a small delay necessary for displaying
* various program messages on the screen.

STORE 0 TO k

DO WHILE k < 100
   STORE k + 1 TO k
ENDDO

RETURN

* eof delay.prg
```

LIST OF REFERENCES

1. Codd, Edward F., "A Relational Model of Data for Large Data Banks", Communications of the ACM, June 1982.

2. Alberte-Hallam, Teresa and Hallam, Stephen F., Micro-computer Use, Academic Press, Inc., San Francisco, California, 1985

3. Senn, James A., Analysis and Design of Information Systems, McGraw-Hill Co., 1984.

4. Kroenke, David, Data Processing (Second Edition), Science Associates, Inc., Chicago, Illinois, 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library Code 0142 Naval Postgraduate School Monterey, California 93843-5002	2
3.	Chief of Naval Operations Director, Information Systems (OP-945) Navy Department Washington, D.C., 20350-2000	1
4.	Computer Technology Programs, Code 37 Naval Postgraduate School Monterey, California 93943-5000	1
5.	Department Chairman, Code 54 Dept. of Administrative Science Monterey, California 93943-5000	1
6.	Professor Norman R. Lyons, Code 54LB Naval Postgraduate School Monterey, California 93943-5000	1
7.	LCDR James R. Duke, Code 54JD Naval Postgraduate School Monterey, California 93943-5000	1
8.	LT Rolando M. Lim SWOSCOLCOM Dept. Head Program Newport, Rhode Island 02840	6

Th
18767 3

LIBRARY
SCHOOL
CALIFORNIA 93943-5002

Thesis
L63865 Lim
c.1 Application of a data-
base system for inventory
management of classified
material.

18 JAN 89 32956
4 APR 89 32893
25 SEP 90 36898

Thesis
L63865 Lim
c.1 Application of a data-
base system for inventory
management of classified
material.

thesL63865
Application of a database system for inv
3 2768 000 75381 8
DUDLEY KNOX LIBRARY